Applicants traverse the rejections of Claims 1-3, 5-9, 12-13, and 18-20, as amended, under 35 U.S.C. § 102(b) as anticipated by Moore et al patent 5,437,017. Reconsideration of the Examiner's position is respectfully requested.

The rejections are based on a misreading of claim language in applying the prior art. There also appears to be a confusion of applicants' "translation buffer" with the "translation lookaside buffer" disclosed in Moore. Applicants system includes both a "translation buffer" and a "translation lookaside buffer." (see Figure 3 in which the "TLB" is illustrated as part of the MMU while the "translation buffer" is shown as part of host memory).

Moore does not teach a computer which includes a host processor designed to execute instructions of a host instruction set and software for translating instructions from a target instruction set to instructions of the host instruction set.

The words "instruction set" have a well known consistently-used meaning in the art. An instruction set is the set of instruction to which the hardware of the processor responds. The pages (previously-submitted in the parent application) from Borland Turbo-Assembler, User's Guide, 1988; Assembly Language, Step by Step, Duntemann, Wiley, 1992; 386SX Microprocessor, Programmers Reference Manual, Intel, 1989; Intel 486 DX Microprocessor, Data Book, 1991; and Intel Pentium Family User's Manual, Vol. 3, 1994, all illustrate the consistent use of this meaning throughout the computer industry.

As a complete reading of the above references will illustrate, an instruction set is different for each particular processor type.

As applicants' specification points out, the invention is used to help translate the instructions of one instruction set into instructions of another instruction set so that a "host microprocessor" capable of executing instructions from one instruction set can execute programs designed for a different processor having a different instruction set.

That this is what applicants teach is illustrated throughout the specification. For example, lines 22-25 on page 6 of the specification state that "The emulator software changes the target instructions of an application program written for a target processor family into host instructions capable of execution by the host microprocessor."

And at page 23, lines 4-11, the specification states:

> More particularly, a morph host is a processor which ... translates the instructions of a target program to morph host instructions for the morph host .... (Emphasis added)

Beginning at page 24, line 24, the specification states:

> The code morphing software combined with the enhanced morph host translates target instructions into instructions for the morph host on the fly and caches those host instructions in a memory data structure (referred to in this specification as a "translation buffer"). The use of a translation buffer to hold translated instructions allows instructions to be recalled without rerunning the lengthy process of determining which primitive instructions are

> required to implement each target instruction, addressing each primitive instruction, fetching each primitive instruction, optimizing the sequence of primitive instructions, allocating assets to each primitive instruction, reordering the primitive instructions, and executing each step of each sequence of primitive instructions involved each time each target instruction is executed. Once a target instruction has been translated, it may be recalled from the translation buffer and executed without the need for any of these myriad of steps. (Emphasis added)

Translating instructions means translating code (not translating addresses) from code for one type of processor to code for a different type of processor. A translation buffer holds translated code (not address translations). Applicants maintain this meaning for the word "translate" consistently throughout the specification.

> The patentee is his own lexicographer. *Canaan Products, Inc. v. Edward Don & Co.*, 156 U.S.P.Q. 295 (7 Cir.); *Eclipse Corp. v. Ford Motor Co.*, 171 U.S.P.Q. 513 (7 Cir.). The terms he uses should be interpreted to read on the structure he describes in the patent. *Canaan Products, Inc. v. Edward Don & Co.*, 156 U.S.P.Q. 295 (7 Cir.). Patentee can choose his own terms and use them as he wishes so long as he remains consistent in their use and makes their meaning reasonably clear. *Eclipse Corp. v. Ford Motor Co.*, 171 U.S.P.Q. 513 (7 Cir.).

The entire invention relates to apparatus and a process by which code which has been translated from target code to host code is protected.

Moore does not translate instructions in one instruction set to instruction in another instruction set. Moore nowhere talks about translating instructions.

Moore describes a computer which uses a plurality of processors all of which may address memory (Abstract, lines 4-5). These processors run instructions in parallel in order to reach higher processing speeds. (col. 1, lines 19-24). In order to run instructions in parallel, each processor must execute the same instruction set since each processor may have to handle any instruction provided by a process. Thus, Moore cannot translate instructions from one instruction set to another instruction set since each processor has the same instruction set.

Each of the claims describes an arrangement in which translated code is protected. For example, Claim 1, as amended, recites:

Each of the claims describes an arrangement in which translated code is protected. For example, Claim 1, as amended, recites:

> A system for maintaining translation consistency in a computer which includes a host processor designed to execute instructions of a host instruction set and software for translating instructions from a target instruction set to instructions of the host instruction set comprising:
>
> hardware means for indicating whether a memory address to be written stores a target instruction which has been translated to at least one host instruction, and
>
> software means responding to an indication that a memory address to be written stores a target instruction which has been translated to at least one host instruction for assuring that host instructions translated from target instructions stored at the

memory address will not be utilized once the memory address has been written.

Claim 12 recites:

> A method of responding to an attempt to write a memory address including a target instruction which has been translated to a host instruction for execution by a host processor including the steps of:
>
> marking a memory address including a target instruction which has been translated to a host instruction,
>
> detecting a memory address which has been marked when an attempt is made to write to the memory address, and
>
> responding to the detection of a memory address which has been marked by protecting a target instruction at the memory address until it has been assured that translations associated with the memory address will not be utilized before being updated.

The other claims have similar limitations.

Moore does not teach the invention claimed in any of these claims. The claims are all directed to an invention in which code intended for a first target processor is translated into code to run on a different host processor which cannot execute the target code. The Moore patent teaches nothing about translation of this sort. For this reason, the rejection of the Claims 1-3 and 5-20 under 35 U.S.C. § 102(b) is respectfully traversed and must be withdrawn.

Since the prior art does not disclose, teach, relates in any way to, or suggest the inventive combination, the withdrawal of the rejection of each of Claims 1-3 5-9, 12-13, and 18-20, as amended, under 35 U.S.C. §102(b) is respectfully requested. Since all claims now in the application appear to be allowable over the rejections included in the Office Action, the allowance of those claims, as amended, and the issuance of the application as a patent are respectfully requested.

Respectfully submitted,

7/15/02

Stephen L. King
Attorney for applicants

4642 Oyster Bay Road NW
Olympia, WA 98502
(360) 867-1379